UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/808,905	03/25/2004	Anson Horton	MS302711.1	7406

27195 7590 01/11/2008

AMIN. TUROCY & CALVIN, LLP
24TH FLOOR, NATIONAL CITY CENTER
1900 EAST NINTH STREET
CLEVELAND, OH 44114

EXAMINER	
SMITH, CHENECA	

ART UNIT	PAPER NUMBER
2192	

NOTIFICATION DATE	DELIVERY MODE
01/11/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket1@thepatentattorneys.com
hholmes@thepatentattorneys.com
osteuball@thepatentattorneys.com

Office Action Summary	Application No.	Applicant(s)	
	10/808,905	HORTON ET AL.	
	Examiner	Art Unit	
	Cheneca P. Smith	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 October 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 March 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on October 31, 2007 has been entered.

2. Claims 1-22 have been examined.

3. Applicant's arguments with respect to claims 1-22 have been considered but are moot in view of the new ground(s) of rejection, as discussed below.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims **1, 5-7, 13-17, and 19-22** rejected under 35 U.S.C. 103(a) as being unpatentable over Bates et al (US Patent Application Publication 2003/0221185 A1) in view of Bates et al (US Patent 7,251,808 B2, referred to as Bates_2, both already of record).

As to claim 1, Bates teaches a computer-implemented attributed debugging

system comprising a debugger that facilitates debugging of a computer software application (see FIG.1: 123) and an expression evaluator (see FIG.1: 126) that evaluates an attribute associated with the computer software application according to an attribute definition, and presents debug information associated with the computer software application in accordance with the attribute definition (see page 6, paragraph [0064]: *at step 614, the debugger determines whether any attributes are set for the variable; if any attributes you set for the variable, then processing proceeds to step 616 where the appropriate attribute indicator (e.g. G,S,I,R,C,P) for each set attribute is associated with the variable value and paragraph [0065]: that is, the debugger determines whether a field is to be displayed*).

Bates does not specifically teach that the debug information is presented in a developer-customizable format. However, in an analogous art, Bates_2 is cited to teach where debug information is presented in a developer-customizable format (see column 2, lines 8-12 and lines 28-37 and FIG. 4 and associated text, e.g. column 3, lines 49-55). It would have been obvious to one having ordinary skill in the art at the time of the invention to combines the teachings of Bates and Bates_2 for the purpose of providing an improved debugger having a mechanism when debugging programs that provides custom record displays, where only user selected fields of records or variables are displayed, as disclosed by Bates_2 (see column 1, lines 53-56).

As to claim 5, Bates in view of Bates_2 teaches the limitations of claim 1, but does not specifically teach that the expression evaluator invokes an overridden implementation of a ToString method to facilitate presentation of debug information.

However, it is well known in the art that ToString methods return a string representation of an object and are used primarily for debugging. It would have been obvious to one having ordinary skill in the art at the time the invention was made to include an overridden implementation of a ToString method for debugging for the convenience of displaying the output for a program and simplifying the process of debugging the program.

As to claim 6, Bates in view of Bates_2 teaches the limitations of claim 1, but does not specifically teach that the expression evaluator employs a result of the overridden ToString method as a value to be displayed for an object, field, property, or combinations thereof. However, it is well known in the art that ToString methods return a string representation of an object and are used primarily for debugging. It would have been obvious to one having ordinary skill in the art at the time the invention was made to include an overridden implementation of a ToString method for debugging and use the returned result as a display value for the convenience of displaying the output for a program and simplifying the process of debugging the program.

As to claim 7, Bates teaches the system of claim 1, the attribute employed to determine at least one of how and whether a type or member is displayed (see page 6, paragraph [0064]: *at step 614, the debugger determines whether any attributes are set for the variable and paragraph [0065]: that is, the debugger determines whether a field is to be displayed*).

As to claim 13, Bates in view of Bates_2 teaches the system of claim 1, the attribute employed to what is displayed for a class and/or field (see Bates: page 6,

paragraph [0064]: *at step 614, the debugger determines whether any attributes are set for the variable and paragraph [0065]: the debugger determines whether a field is to be displayed*).

As to claim 14, Bates in view of Bates_2 teaches the system of claim 13, an argument to the attribute comprising a string that is displayed in a value column for an instance of the class and/or field (see Bates: page 6, paragraph [0064]: *if any attributes are set for the variable, then processing proceeds to step 616 where the appropriate attribute indicator is associated with the variable value*).

As to claim 15, Bates in view of Bates_2 teaches the system of claim 14, the argument associated with one of a field, a property or a method (see Bates: page 6, paragraph [0065]: *the debugger determines whether the variable value is associated with a field of a record*).

As to claim 16, Bates in view of Bates_2 teaches the system of claim 11 further comprising an attribute cache directory (see Bates: FIG. 1: 150) that stores an attribute associated with the computer software application, the expression evaluator employing the stored attribute to present debug information (see Bates: FIG. 1: 150 and page 3, paragraph [0033]: *the database management system includes a database which may be a variety of repositories... the database provides one example of an external data source for external comments and other variable information*).

As to claim 17, Bates teaches a computer-implemented method facilitating attributed debugging comprising determining whether a process has an attribute attached (see page 6, paragraph [0064]: *at step 614, the debugger determines whether*

any attributes are set for the variable) and utilizing a definition of the attribute to display debug information, if the process has an attribute attached (see paragraph [0065]: *the debugger determines whether a field is to be displayed*).

Bates does not specifically teach that the debug information is displayed in a developer-customizable format. However, in an analogous art, Bates_2 is cited to teach where debug information is presented in a developer-customizable format (see column 2, lines 8-12 and lines 28-37 and FIG. 4 and associated text, e.g. column 3, lines 49-55). It would have been obvious to one having ordinary skill in the art at the time of the invention to combines the teachings of Bates and Bates_2 for the purpose of providing an improved debugger having a mechanism when debugging programs that provides custom record displays, where only user selected fields of records or variables are displayed, as disclosed by Bates_2 (see column 1, lines 53-56).

As to claim 19, Bates in view of Bates_2 teaches a computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 17 (see FIG. 1 and associated text).

As to claim 20, Bates teaches a data packet stored on computer readable media, the data packet transmitted between two or more computer components that facilitates debugging, the data packet comprising an attribute, the attribute providing information associated debugging of a computer software application (see page 4, paragraph [0047]: *fields 312-322 are flags whose value describes an attribute of the variable*).

Bates does not specifically teach that the information is displayed in a developer-customizable format. However, in an analogous art, Bates_2 is cited to teach where

information is presented in a developer-customizable format (see column 2, lines 8-12 and lines 28-37 and FIG. 4 and associated text, e.g. column 3, lines 49-55). It would have been obvious to one having ordinary skill in the art at the time of the invention to combines the teachings of Bates and Bates_2 for the purpose of providing an improved debugger having a mechanism when debugging programs that provides custom record displays, where only user selected fields of records or variables are displayed, as disclosed by Bates_2 (see column 1, lines 53-56).

As to claim 21, Bates teaches a computer readable medium storing computer executable components of an attributed debugging system comprising: a debugger component that facilitates debugging of a computer software application (see FIG.1: 123); and an expression evaluator (see FIG.1: 126) component that employs an attribute definition to evaluate an attribute associated with the computer software application to present debug information associated with the computer software application to a developer (see page 6, paragraph [0064]: *at step 614, the debugger determines whether any attributes are set for the variable; if any attributes you set for the variable, then processing proceeds to step 616 where the appropriate attribute indicator (e.g. G,S,I,R,C,P) for each set attribute is associated with the variable value and paragraph [0065]: that is, the debugger determines whether a field is to be displayed*).

Bates does not specifically teach that the debug information is displayed in a developer-customizable format. However, in an analogous art, Bates_2 is cited to teach where the debug information is presented in a developer-customizable format (see

column 2, lines 8-12 and lines 28-37 and FIG. 4 and associated text, e.g. column 3, lines 49-55). It would have been obvious to one having ordinary skill in the art at the time of the invention to combines the teachings of Bates and Bates_2 for the purpose of providing an improved debugger having a mechanism when debugging programs that provides custom record displays, where only user selected fields of records or variables are displayed, as disclosed by Bates_2 (see column 1, lines 53-56).

As to claim 22, Bates teaches a computer implemented attributed debugging system comprising: means for storing an attribute associated with a computer software application (see page 3, paragraph [0033]: *the database management system includes a database which may be a variety of repositories... the database provides one example of an external data source for external comments and other variable information*) and means for employing the stored attribute and an attribute definition to present debug information associated with the computer software application to a developer (see page 6, paragraph [0064]: *at step 614, the debugger determines whether any attributes are set for the variable; if any attributes you set for the variable, then processing proceeds to step 616 where the appropriate attribute indicator (e.g. G,S,I,R,C,P) for each set attribute is associated with the variable value and paragraph [0065]: that is, the debugger determines whether a field is to be displayed*).

Bates does not specifically teach that the debug information is displayed in a developer-customizable format. However, in an analogous art, Bates_2 is cited to teach where the debug information is presented in a developer-customizable format (see column 2, lines 8-12 and lines 28-37 and FIG. 4 and associated text, e.g. column 3,

lines 49-55). It would have been obvious to one having ordinary skill in the art at the time of the invention to combines the teachings of Bates and Bates_2 for the purpose of providing an improved debugger having a mechanism when debugging programs that provides custom record displays, where only user selected fields of records or variables are displayed, as disclosed by Bates_2 (see column 1, lines 53-56).

6. Claims **2-4 and 8-12** are rejected under 35 U.S.C. 103(a) as being unpatentable over Bates et al (US Patent Application Publication 2003/0221185 A1) in view of Bates et al (US Patent 7,251,808 B2, referred to as Bates_2) as applied to claims 1 and 7 above, and further in view of Dandoy (US Patent Application Publication 2004/0230954 A1).

As to claim 2, Bates in view of Bates_2 teaches the limitations of claim 1, but does not specifically teach that the expression evaluator evaluates an expression associated with a particular programming language. However, in an analogous art, Dandoy teaches a system for debugging a software application that displays the contents and properties of objects and determines what events are emitted by objects (see paragraph [0046]). It would have been obvious to one having ordinary skill in the art to combine the teachings of Bates and Bates_2 with that of Dandoy because the systems and methods of Dandoy's invention can be configured to debug software that was created with other programming languages besides Java or different variants of Java (see page 6, paragraph [0048]).

As to claim 3, Bates teaches a debugger and an expression evaluator, but does not specifically teach the programming language is at least one of C#, J# or Visual Basic.Net. However, in an analogous art, Dandoy teaches a system for debugging a software application that displays the contents and properties of objects and determines what events are emitted by objects (see paragraph [0046]). It would have been obvious to one having ordinary skill in the art to combine the teachings of Bates and Bates_2 with that of Dandoy because the systems and methods of Dandoy's invention can be configured to debug software that was created with other programming languages, including Java, HTML, C#, C++, and C (see page 6, paragraphs [0048] and [0049]).

As to claim 4, Bates teaches a debugger and an expression evaluator, but does not specifically teach a plurality of expression evaluators, each expression evaluator associated with a particular programming language. However, in an analogous art, Dandoy teaches a system for debugging a software application that displays the contents and properties of objects and determines what events are emitted by objects (see paragraph [0046]). It would have been obvious to one having ordinary skill in the art to combine the teachings of Bates and Bates_2 with that of Dandoy because the systems and methods of Dandoy's invention can be configured to debug software that was created with other programming languages, including Java, HTML, C#, C++, and C (see page 6, paragraphs [0048] and [0049]).

As to claim 8, Bates teaches the limitations of claim 7, but does not specifically teach the attribute employing an enumeration. However in an analogous art, Dandoy teaches how the debug agent is configured to collect execution data relating to the

graphical user interface, which includes object properties, events, and runtime states (see page 2, paragraph [0018]). It would have been obvious to one having ordinary skill in the art at the time of the invention to combine the teachings of Bates and Bates_2 with that of Dandoy for the advantage of gaining a more efficient debugging system that does not require the source code of the application being debugged to be modified and saving programmers and developers valuable time.

As to claim 9, Dandoy further teaches one enumeration value associated with an indication that the type or member should not be displayed to the developer (see page 3, paragraph [0025], *the debug agent can determine the current state of the selected window and change its properties such that the window is hidden*).

As to claim 10, Dandoy further teaches one enumeration value associated with an indication that if a type is hierarchical, it should be expanded by default (see page 5, paragraph [0046]: *at any point during debugging, a hierarchy of objects within the interface can be determined and displayed; the hierarchy can be displayed automatically*).

As to claim 11, Dandoy further teaches one enumeration value associated with an indication that a type should not be expanded by default (see page 3, paragraph [0024], *debugging requests may include a request to monitor events associated with an object... or request to hide or show an object*).

As to claim 12, Dandoy further teaches one enumeration value associated with an indication that a target element itself should not be shown, but should instead be automatically expanded to have its member(s) displayed (see page 5, paragraph [0046],

at any point during debugging, a hierarchy of objects within the interface can be determined and displayed; the hierarchy can be displayed automatically).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cheneca P. Smith whose telephone number is (571) 270-1651. The examiner can normally be reached on Monday-Friday 7:00-4:30 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

CS
1/4/2008

TUAN DAM
SUPERVISORY PATENT EXAMINER